

**mRNA: A Digital Annotation System to
Facilitate Multi-disciplinary Group
Collaboration**

by

Pedram Keyani

Department of Computer Science

Stanford University

Advisors

Terry Winograd

Andreas Paepcke

mRNA: A Digital Annotation System to Facilitate Multi-disciplinary Group Collaboration

by

Pedram Keyani

Abstract

Communication overhead can be detrimental to the success of group collaboration involving a large multi-disciplinary team working over an extended period of time. Large groups incur the communication overhead of organizing, reading and filtering large volumes of emails based on whom in the organization sent them. Multi-disciplinary teams must overcome terminology differences to work together. Collaborative efforts spanning extended periods of time also tax the limits of human memory. The mRNA project studies the use of digital annotations on a shared corpus as an efficient lightweight means of communication. The goal is to minimize communication overhead while maximizing the amount of relevant information that participants receive. The work has been focused on analyzing CSCW and annotation research to identify interesting areas of future research. This project has generated new research ideas, preliminary design of an annotation framework and a prototype annotation client.

Contents

1 Introduction.....	4
2 Problem Statement.....	5
3 Background.....	6
4 Related Work.....	8
4.1 CSCW.....	8
4.1.1 WikiWeb.....	9
4.2 Digital Annotation Systems and Frameworks.....	11
4.2.1 Multivalent Documents (MVD).....	11
4.2.2 Commentor.....	13
4.2.3 Microsoft Annotation Tools.....	14
4.2.4 Common Annotation Framework.....	17
4.3 Annotation Systems in the Sciences.....	18
4.4 Knowledge Representation.....	18
4.4.1 A KR is a Surrogate.....	19
4.4.2 A Set of Ontological Commitments.....	19
4.4.3 A Medium for Efficient Computation.....	19
4.4.4 A Medium of Human Expression.....	20
5 Approach.....	20
5.1 Reading of Documents Should Be Natural.....	21
5.2 Reading and Writing of Annotations Should Be Simple.....	22
5.3 Important Annotation Anchors.....	23
5.3.1 Anchoring to Documents.....	24
5.3.2 Anchoring to Segments of Text.....	24
5.3.3 Anchoring to Specific Terms.....	24
5.4 Avoid Information Overload.....	25
6 System Architecture.....	28
6.1 mRNA Browser.....	29
6.2 mRNA Annotation Server.....	30
7 Conclusion.....	31
7.1 Annotation Architectures.....	31
7.2 Annotations in Supporting Group Collaboration.....	32
7.3 Annotations as a communication Medium.....	33
7.4 Lessons Learned About HCI Research.....	33
8 Future Work.....	34
A Note About the Name mRNA.....	36
Acknowledgements.....	37
Bibliography.....	38

1 Introduction

Complex problems often require large multi-disciplinary teams to solve them. Examples can be found in the field of bioinformatics and gene-therapy research where geneticists, chemists, computer scientists and other domain experts work together to tackle problems that no one field can deal with alone. Unfortunately, group collaboration incurs a heavy communication overhead to coordinate between members and share knowledge about the problem solution. Lack of communication can lead to duplicated work, stress and erroneous results. The problems of collaboration are compounded when members of the group have different frames of reference or disjoint sets of knowledge. Along with verbal communication, documents can be used to share, clarify and save ideas.

A typical workgroup will often create, store, destroy, read and share digital documents in the course of their collaboration. Digital documents come in the form of web pages, text documents, graphs, tabular data and any assortment of formatted data. Regardless of their representation they can store magnitudes of information, of which a small subset may be of relevance to any one person in a team.

This project is aimed at minimizing communication overhead and increasing shared knowledge in multi-disciplinary teams. I extended many ideas in Computer Supported Collaborative Work (CSCW) and annotation research to develop new ideas for using annotations as a communication medium on a shared corpus. The challenge in this type of work is in reducing information overload while increasing shared knowledge. My approach to handling information overload is by creating a set of filters both at the authoring and reading stage. Increasing shared knowledge is less straightforward; my

solution is to create three different types of annotation anchors (document, free text and term) that vary how annotations are replicated. The anchors serve as a mechanism for replicating annotations and distributing shared comments. The details of this work will be described in much greater detail later in this report.

Section 2 will give a deeper account of the problem and describe supporting examples in the field of scientific research. Section 3 will give an overview of conventional paper-based annotations and studies to determine how they are used effectively. Section 4 will describe related work in digital annotation systems and describe their unique contributions. Section 5 will describe my solution in more detail. Section 6 will describe the system architecture. Section 7 focuses on lessons learned and anecdotal results. Finally, section 8 outlines future directions for this research.

2 Problem Statement

My research is aimed at creating a communication infrastructure for multi-disciplinary, scientific, collaborative researchers working for long periods of time. There are too many variables to tackle all at once; table 1 below lists the major components and the problems that ensue.

What we want to support	Problems to overcome
Collaboration	Team size and location determine what forms of communication are effective and feasible.
Multi-disciplinary teams	Communication between different types of domain specialists can be slow due to different frames of reference and different meanings used for similar words.
Research	Research, in general, deals with large volumes of documents. The initial investigative stage requires reading many documents while the latter experimentation or simulation stage often produces large amounts of data. Scientific data often requires special types of visualization and specialized viewers to access the data.

Long time span	Human memory is a limiting factor for projects that deal with large amounts of data for a long period of time.
Combination of all the above	Large volumes of documents are produced and shared over a long period of time and sorting through and comprehension of it all can be a daunting if not impossible task.

Table 1. Different components of the problem

A motivating example that drives this work is in the life sciences, specifically cancer drug research. This is a problem that requires the specialization of chemists, geneticists, bioinformaticists, and computer scientists to name a few. Their goal is to develop drugs that fight cancer and each member of the team has a different conception of how to get there and a different set of skills that can be applied to the problem. Their work may involve drug development, laboratory experiments, user studies, extensive reading, modeling and simulation; all of which create a magnitude of data. In the process of moving towards their goal the group's collaboration and communication centers on this data. It is in this potential sea of data that chaos emerges.

3 Background

Annotating comes naturally to most of us. From an early age we learn to fold the corner of important pages, known as dog-earing, and with more complicated information we often underline or write in the margins, known as marginalia. These marks and comments serve as visual cues to distinguish important sections of text and a means to store insight about the content. While these capabilities of annotations for an individual have long been understood, more recently annotations have gained attention as a form of communication between people working on a shared corpus [MDL].

A very notable research project performed detailed analysis on how students annotate textbooks in order to understand the different annotation types and usage patterns [ECO]. This research was part of an effort to understand annotations and hypertext authoring systems as they relate to digital libraries of the future [MAR]. The work describes annotations on many dimensions; table 2 lists the dimensions relevant to the mRNA project.

Dimension	Explanation
Formal vs. informal	On the formal end is Metadata with free form text on the informal end. This equates to how effective the annotation is for automatic indexing, storage and searching.
Annotation as writing vs. reading	The level of interaction that the annotation author has with augmenting/changing the form and content of the original document.
Permanent vs. transient	This captures the temporal aspect of the annotation as it relates to individual and group interaction with the document. Not all annotations are created equal and some are appropriate only for a short period of time.
Published vs. private	Just as some documents are meant to be private, not all annotations are meant for public viewing.
Global vs. institutional vs. workgroup vs. personal	Hypertext authoring, as it exists on the Internet, is an example of a global system while notes in a personal journal are just that, personal. Our interest lies in the institutional and workgroup level annotations and that is an area of exploration for this work.

Table 2 Dimensions of annotations

Using annotations as a form of communication is more feasible with digital documents where information can be replicated and transferred automatically. While it offers many powerful tools for communication, many of the ideas must be developed and tested out in the real world. My aim is to apply the lessons learned from this research

project and ideas from other digital annotation systems to develop design principles for a collaborative annotation framework.

4 Related Work

This section describes Computer Supported Collaborative Work (CSCW), digital annotation systems and data engineering. It is at the intersection of these fields that interesting new collaborative annotation technologies will emerge.

4.1 CSCW

The field of Computer Supported Collaborative Work is rather large with a wide array of projects working in many different directions that are unified by the use of computer technology to tackle the social and cognitive constraints of group collaboration. Work by [CSC, SER] classifies CSCW into low-level and high-level facilities. Low-level facilities cover collaborative editing, annotations and other forms of basic communication. High-level refers to organizational process and group interaction. Their focus is on the effective application of low-level facilities to smoothly integrate into the high-level work practices of groups. They point out that CSCW technologies must be integrated into the tools that people commonly use in order to really prove useful. Interesting CSCW work on products and standards has been done at Microsoft Research to address that issue [NOT, CAF, AAC]. Some of this work has made its way into commercial application such as the “web discussions” feature of Office 2000.

4.1.1 WikiWeb

The Wiki concept is an extreme example of a quick, ubiquitous and easy to use collaboration tool, many of the other technologies I discuss will move away from this simple lightweight form of collaboration. The WikiWikiWeb server concept, or Wiki, is a way for groups of people to collaborate simply and quickly on the web [WIK]. The word Wiki comes from the Hawaiian word for quick, “WikiWiki”, and it captures the main concept of making web content easily editable by anyone.

The basic design is shown in figure 1. The design centers around a shared set of pages accessible and editable by multiple users on the Internet. Each user has the ability to view, edit, link and create pages using any web browser that supports forms. At the bottom of each Wiki page is a link “Edit Text”; selecting it will bring the user to a text box that allows the user to edit the page. Adding or editing content is as simple as altering the text. The really elegant but powerful capability is in the simple way of creating links and new pages within the wiki. In order to create a link you simply string two uppercase words together such as “DigitalAnnotation”, this will either create a link to an already existing page within the wiki or create a new empty page “DigitalAnnotation”. The concept is very simple, making it easy for beginners and yet fast for power users.

The original Wiki implementation was in Perl with a wide variety of clones to follow such as TWiki, SWiki or CoWeb, Pyke, Rwiki, Zwiki, and a many other subtle variations. Two Wiki clones of interest to my work are TWiki and ZWiki. TWiki is meant for corporate use and supports revision control, email notification and templates. ZWiki is written in Python and integrated with Zope which is a object publishing

environment [ZOP]. Twiki makes use of XML to encode meta-data and allow for many data types to be published and accessed in a networked environment [TWI].

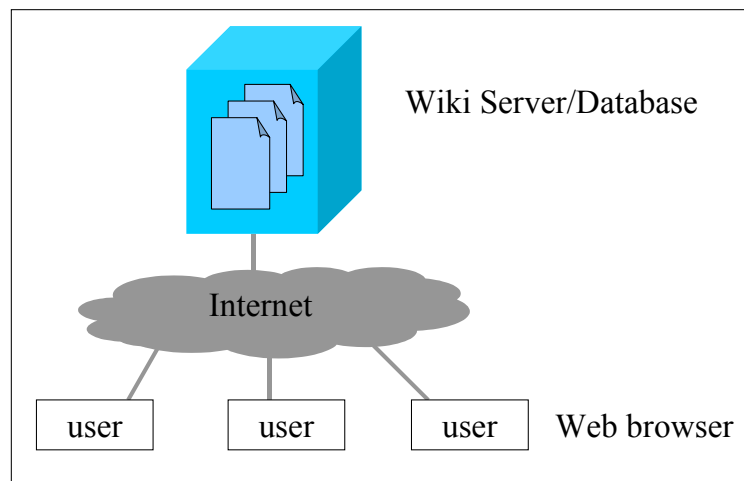


Figure 1: Basic wiki structure

Wikis have been used successfully in educational and business settings to effectively communicate and collaborate in tightly focused or small groups. Successful examples can be found at Georgia Tech, New York Times Digital, TakeFive software (now WindRiver), and other companies and organizations that have used wikis and Wiki clones for a variety of purposes.

The simple Wiki concept has been evolving since 1994, while staying true to its roots of fast and simple authoring of shared online material. It has been deployed in academic, recreational and professional settings. The main lesson that these cases show is that for a wiki to be utilized, the group must either be of moderate size, with an upper limit of about 20 as observed by Motorola, or be tightly focused on a particular subject [WIK]. The reason for this is that content and organization are not moderated and as such, the wiki quickly grows unwieldy to traverse if there are too many topics or too many opinions. While search capabilities and visualization tools can alleviate this to some extent, there is a practical limit where these tools can no longer help. Extensions

such as ZWiki and TWiki provide practical capabilities that can extend the Wiki concept to other usage scenarios.

4.2 Digital Annotation Systems and Frameworks

Digital annotation research has been underway for a number of years and gained much interest with the rise of the Internet and the newfound interest in digital libraries. Many of the projects described in this section are sizable undertakings and have overlaps with each other so I have focused on the important aspects of each as they relate to my work. A more detailed survey of many digital annotation systems can be found in [SYS]. Their coverage focuses on the underlying design decisions to support digital annotation systems.

4.2.1 Multivalent Documents (MVD)

MVD is a research initiative at Berkeley that is working to bring the flexible manipulation affordances of paper to digital documents [MVD]. They have designed an extensible annotation framework that extends the ease of digital document editing, storage, distribution and indexing to annotations. Their framework is based around the notion that effective digital annotations should appear *in situ*, be format independent, leave the original document unaltered, and be robust to changes. They have developed a novel layered approach to annotation composition as well as an architecture for network portable annotations.

The MVD annotation model has three types of annotations: span of elements, geometric region and tree structure. These types are differentiated by how the annotations are anchored and presented. The details of each are less important than the

fact that they identified the need for putting annotations into different categories. We will revisit this point when discussing my annotation categorizations. Another interesting point is that they regard documents as compositions of related but distinct layers and dynamically loaded behaviors. In other words, the original document constitutes one layer and different annotation types form other layers that are used to decorate this base information layer.

The networked MVD architecture centers on a proxy server that works in concert with MVD viewers to combine the original documents and annotations to form the document the user sees. Figure 2 shows a high level view of the major components. The process starts when a person enters the URL of a document into an MVD browser and it is relayed to the MVD proxy server that handles the request. The proxy server will then attempt to locate the document and add an annotation layer on top of it. The resulting document is returned to the client browser in its new form. The project has also defined a set of protocols that behaviors conform to, thus allowing third party developers to create their own MVD compliant behaviors.

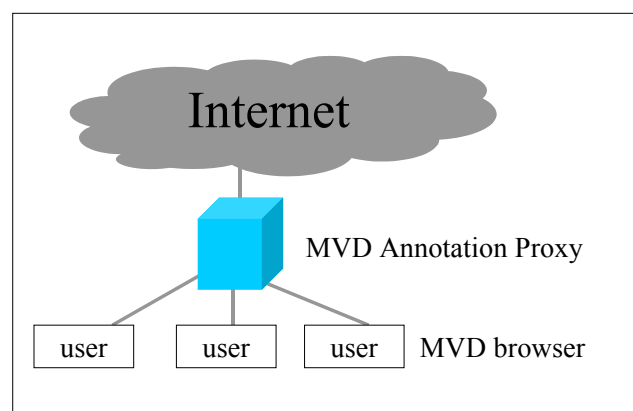


Figure 2. Basic MVD architecture

4.2.2 Commentor

The Commentor architecture was created at Stanford University to support synthesis of documents and third-party meta-data without affecting the original content [COM]. Many of the useful scenarios that it was designed for involve group collaboration and communication. The Commentor architecture is similar to the MVD model in that meta-data is kept separate from the original documents, but differs in two significant ways. First, it does not rely on a proxy to deliver the original documents. Second, it accounts for group work and organizational structure. The Commentor architecture as shown in figure 3 separates the storage and representation of meta-information from the underlying documents. Their specialized browser makes requests to the document server in order to get the original document and any annotations are retrieved from the individual meta-information servers. There can be a number of meta-information servers, each performing a different service or for a different group. The Commentor architecture and protocol define the specific interactions that the browser has with the different servers but the last step is for the browser to combine and synthesize all this information.

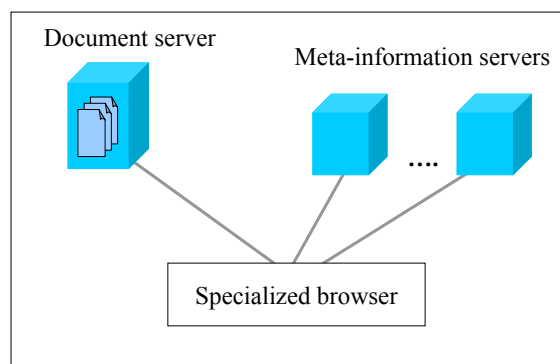


Figure 3. Commentor architecture

Another significant contribution of Commentor is the recognition of group collaboration as a central usage of annotations. As a result of this, they have devised an authentication,

authorization and selection scheme as shown in figure 4. This model allows users to belong to multiple groups who in turn hold multiple annotation sets with varying views. The key to this model is that it allows for organizational structuring without being overly specific as to be tied to one type of organizational structure. While the notions of user, group and view are intuitive to understand, sets are less intuitive. I will give a more detailed discussion of this as it is the motivation for my context based filtering. Table 3 lists the more compelling usage scenarios described in [COM].

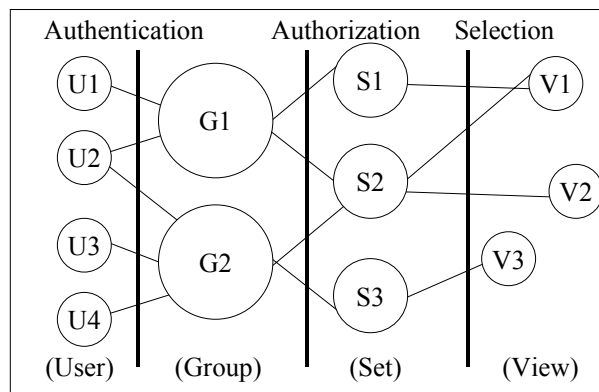


Figure 4 Group structure as reflected in design

Supported concept	Description
Structured discussions	Using annotations as a way to communicate about a document
Seal of approval (SOAP)	Rating system for documents
Landmarks, Tours and Trails	Ways to navigate large webs of documents
Usage indicators	Monitoring how often documents are viewed or edited

Table 3. Possible uses of the Commentor architecture

4.2.3 Microsoft Annotation Tools

Microsoft is one of the first major corporations to provide annotation support for their applications. Office 2000 is one of the first widely used commercially available tools to provide collaborative annotation tools. Studies were performed to assess how the system was used over a long period of time and their findings are very positive for using

annotations as an effective asynchronous communication medium. Another Microsoft experiment in conjunction with the University of Washington explored the use of notifications with annotations to improve shared awareness.

Cadiz et al. performed a sizable in-house study to observe how a software development team made use of the system for online software specifications [ACC]. Their study consisted of 450 members making 9,239 annotations on 1,243 documents over a ten-month period of time. The participants were part of a software development team where members were classified as program manager, tester or developer. The team used the “web discussions” feature of Office to replace their former paper-based method of evaluating and commenting on software specifications. The research team collected quantitative data by way of the annotation server logs and qualitative data through user interviews.

Once the study was complete, the research team copied the contents of the annotation server to analyze the results. They categorized users as heavy users, light users and one-time users; they found that 32% of the users made 79.6% of the annotations. They made an interesting observation that program managers, testers and developers produced a comparable amount of annotations. They also observed that there was a high variability in the amount of annotations that were made to particular documents and the time between the first and last annotations made to documents. This may stem from the fact that not all documents are equally important. This variability was handled by the email notification mechanism they implemented. Users could subscribe to notifications on a per-document level based on specific changes daily updates or weekly updates. 269 of the users subscribed to notifications of one document

or more and 70% of these notifications were on a daily basis. A sizable chunk, 28%, of the notifications were set on a per-change basis, no further information was given as to the nature of those documents.

User interviews were used to try and understand some of the numbers that the team observed. They observed that some of the one-time users stopped using the system because changes to the underlying text would orphan their annotation. Proper annotation anchoring is an important aspect to an effective annotation system to keep people using the system. Beyond the simple issue of proper anchoring is the problem of the annotation getting out of synch with the text, as it is edited, causing confusion and even frustration among the users.

They found that users did not use the system to make high-level comments or, what could be interpreted as harsh remarks, due to the public nature of the annotations. Another finding was that users were hesitant to use the annotations to communicate on time-critical matters. This common perception was backed up by figures that showed that annotation response time took on average 7 days. They concluded that the system could be made more effective if the notification mechanism were coupled with another lightweight communication application such as MSN or AOL messenger.

They maintained the email system as a subscription-based notification mechanism but made changes to how it displayed information. Since the details are minimally important to this project I will defer further discussion of this to [NOT].

The feedback was used to revamp the email notification system and combine the system with MSN Sideshow, which is a peripheral awareness tool. This system was deployed in a field study on another software development project to test the

effectiveness of the changes [NOT]. Sideshow is a small sidebar on the desktop that uses the notion of tickets to provide peripheral information on a wide variety of event types. It is a more lightweight notification mechanism than email, and the results show that many of the participants felt that it kept them more up to date on documents and allowed them to respond more rapidly than email notifications.

They also found that notification effectiveness partly depends on the number of documents a user deals with and the rate at which they change. In particular, people responsible for a small number of documents found marginal benefit in using Sideshow. Another finding was that users viewed the ease of configuring their notifications as a critical aspect of the system. Below is a list of necessary features that have come from their study for an effective annotation notification system.

- Keep users aware of annotations activity around documents they are interested in
- Send users the right amount and type of information at the right time
- Support a lightweight subscription mechanism and be easily customizable
- Operate via familiar and convenient channels of communication
- Provide a means for users to easily follow-up on annotation activity
- Provide meta awareness of who is subscribed for notification on a document
- Provide a means for annotators to change notification frequency or modify who will be notified

4.2.4 Common Annotation Framework

CAF is a Microsoft initiative to easily support annotations across applications [CAF]. It is an effort to increase interoperability among annotation-enabled applications. While many frameworks such as the Resource Description Framework (RDF) exist, Microsoft chose to develop and spread CAF. Their design goals were to create a simple, extensible, storage-neutral, universally supported, annotation framework. Currently commercial annotation systems are not widespread enough for one standard to prevail but because of Microsoft's backing, CAF is well poised regardless of technical/design merit.

4.3 Annotation Systems in the Sciences

Annotation systems have found their way into fields such as genome research and bioinformatics. These fields have many nuances that general-purpose computer users rarely face. Distributed Annotation System (DAS) attempt to integrate all the different disparate data sources from different laboratories for genome data [DAS]. The Biological Naming Service (BNS) work at Agilent resolves different biological names and provides many properties of biological data including annotations [BNS]. The way that this data/knowledge is represented determines how it is used and how easily different organizations can join forces to collaborate and solve difficult problems. Inter-organizational and even intra-organizational scientific collaboration often deals with problems of different notations, naming conventions and data representations.

4.4 Knowledge Representation

Knowledge representation (KR) is a central field of AI, but its concepts have been applied to the Internet, collaboration and annotation of objects [W3C]. Classical AI research in this field is centered on how to represent knowledge of the world or some special situation so that a computer can reason about the situation. In 1993 Randall et al. shared their views on KR as a response to the many disparate notions of how to represent knowledge in the AI community [KRR]. They advocated viewing KR as having 5 different roles, when thinking about representation. In the following subsections I will summarize the roles that make sense in relationship to multi-disciplinary collaboration. While their point of view is meant to guide KR research I am drawing from their lessons to facilitate group collaboration.

4.4.1 A KR is a Surrogate

Intelligent beings reason inside co their minds about both abstract and concrete notions of the world. They must store knowledge of the world internally and this raises questions of how well or correctly the external entity is being represented. An important extension of this original question is how to ensure that the different mental surrogates of a collaborating team match up. Misconceptions can be costly to any collaborative effort.

4.4.2 A Set of Ontological Commitments

Randall et al's. view of KR is that any representation that is chosen will have tradeoffs in what it can and cannot represent accurately. In fact, they take the stance that all representations are flawed approximations of the real or abstract object. The approximation determines what we recognize and what we ignore. This is true of any field where models of real systems are built to reason about them. In order for the model to be effective for a multi-disciplinary team it must be able to capture all the relevant points for each team member while allowing each member to selectively ignore superfluous information.

4.4.3 A Medium for Efficient Computation

This role deals with the computational efficiency by which a KR allows a system to reason. This raises the question of how much annotation information should be represented in a human readable way. The answer really lies in how much automation one needs in an annotation system. While conveying annotations in a completely computer readable format provides many opportunities for automation, it also makes it

more difficult for the user to annotate quickly and naturally [ECO]. The Commentor system makes extensive use of meta-data to allow complex filtering and searching.

4.4.4 A Medium of Human Expression

A KR is also a means by which we communicate to the computer and others. A KR design intended to support human communication must be expressive enough to be meaningful to the users. For collaborative systems the designers must determine how the participants will communicate and devise their system to allow the maximum level of expressiveness needed with the minimal effort to the users. The question of how exactly to do this can only be answered by studying the forms of communication within a workgroup and determining how expressive it is.

This related work in annotations, collaboration and knowledge representation resonates with the idea that a shared notion of reasoning, understanding and expression is the key to effective collaborative work. Of the systems that were analyzed, there is no clear choice as to which system is best, as each system makes tradeoffs depending on what it was built for. It is clear that since Microsoft's work has already made its way into consumer applications, their notions of collaboration have a strong chance of becoming a de-facto model for group-based annotation.

5 Approach

In this section I will describe my proposed solution and how my design decisions attempt to address the problems of a multi-disciplinary collaborative setting. In the course of my research I built a prototype annotation-enabled web browser, figure 5, to implement my

ideas. The system architecture will be described in further detail in section 6. In my approach to the problem I am focusing on the type of annotation where a user selects a section of text and adds a textual note to it. The reason I focus on this model of annotation is that it is the most effective form of using annotations as a communication medium. While my system can be used for other types of annotations such as simply underlying text or highlighting passages, this is not the intended usage scenario. This is because highlights and underlines are not expressive enough to effectively communicate relevant facts to users who may not have time to read the entire document to understand the full context.

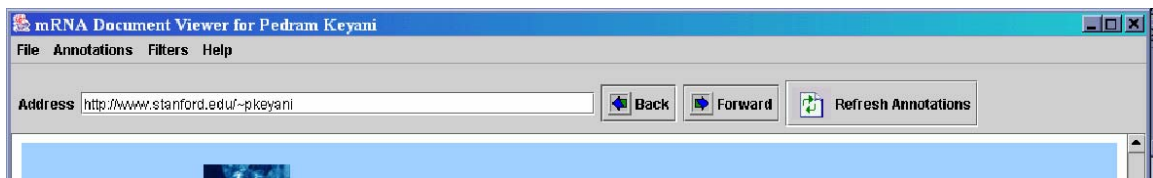


Figure 5. Prototype browser

5.1 Reading of Documents Should Be Natural

In an effort to build an annotation system I was very conscientious about keeping the primary task of reading documents unchanged. In designing the system I did not want to put annotation body in line with the text because the user would have to actively filter out unwanted information. My approach is to highlight text that has been annotated and users can double-click on the highlight to see the annotation. Figure 6 shows the highlighted text on the left and the annotation window that the user sees when selecting the annotation. Notice that annotations from multiple users are put in the same window, I will discuss how this relates to filtering in section 5.3

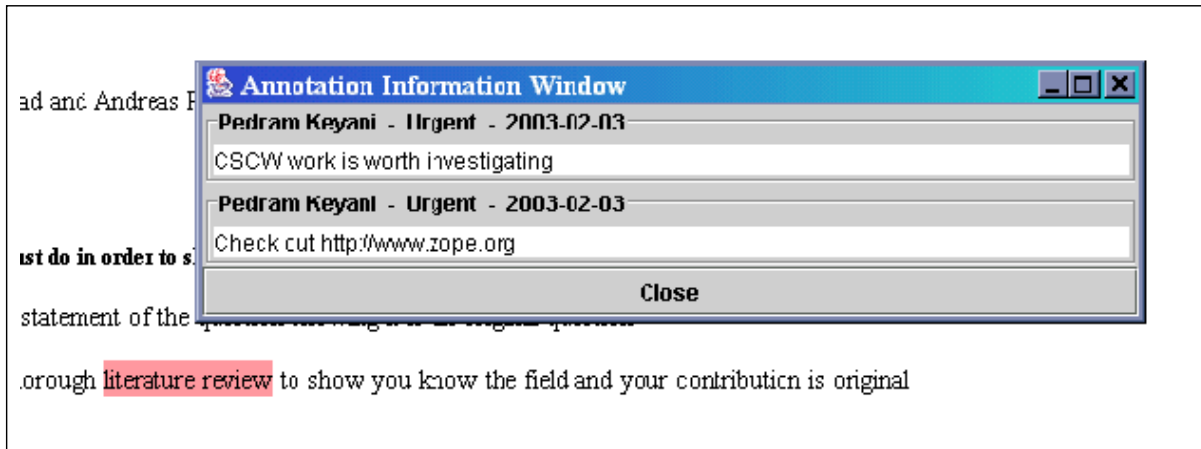


Figure 6 Screen shot of highlighted text on left and annotation window on right

Almost as important as the content of the annotation is the visual cue used to direct users to important annotations. My approach was to color code the annotations in terms of their importance, this is shown in Figure 7 a. I wanted my UI to be as flexible to user preferences as possible so I put in features to allow color intensity and highlight style to be changed as well, this is shown in figure 7 b.

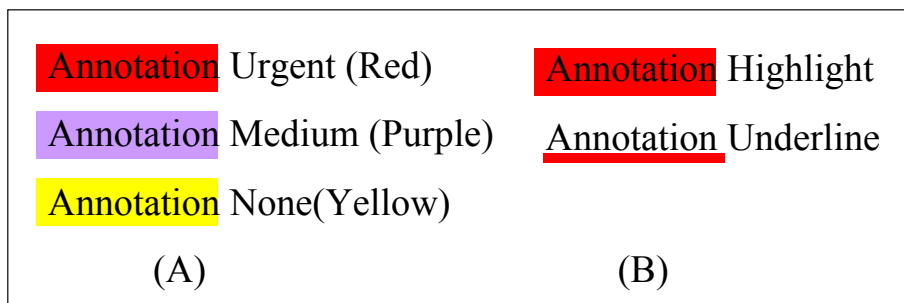


Figure 7 Different representations of annotations. (A) Colors (B) Types

5.2 Reading and Writing of Annotations Should Be Simple

My guiding principle was that reading and writing of annotations must be simple activities for the users. If the process of reading and writing annotations is costly in terms of brainpower exerted and time spent, then it defeats the central purpose of the project. Figure 8 shows the three steps involved with creating an annotation. The first

step is to select the section of text, then the user selects to create an annotation by right clicking to bring up a popup menu and finally the user enters the annotation options that they want and save the annotation. I will focus on the details of annotation creation window in sections 5.3 and 5.4, now I will focus on using annotation templates to streamline the annotation process.

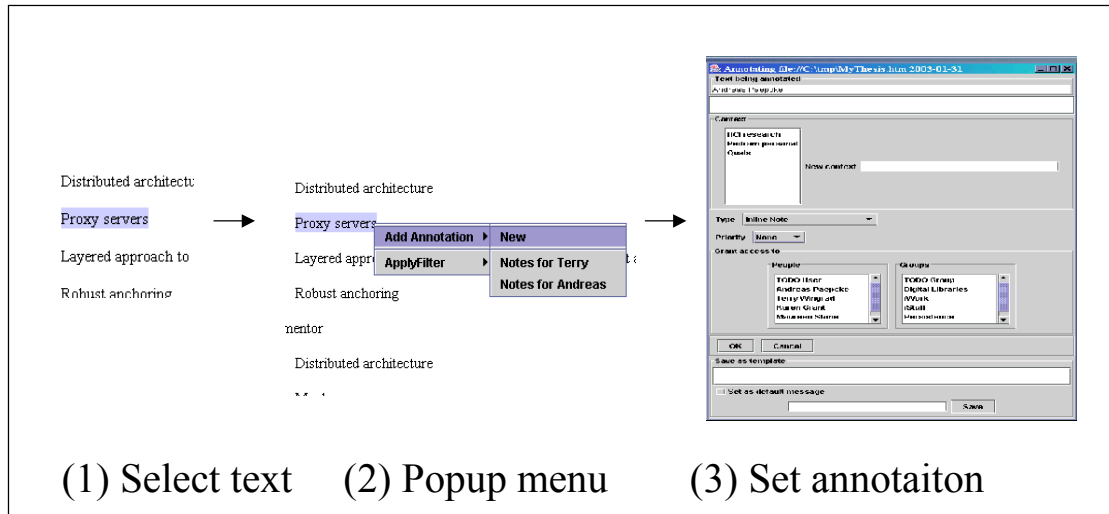


Figure 8. The steps in creating an annotation

In Figure 8 the user has the option to also save the values of the annotation into a named annotation template. If the user fills in the bottom text field and saves that as the default text then annotation writing is as simple as selecting text, right clicking and selecting the name of the annotation template that they chose. This type of shortcut will prove to be very useful for users who write the same types of notes many times. An example would be an urgent annotation from a user that always said “google for” or “recheck results”.

5.3 Important Annotation Anchors

In my research I have identified three essential types of annotation anchors. To summarize, an anchor is what the annotation references, in layman’s terms, what it sticks

to. One of the most powerful aspects of digital media is the ease of replication and my aim was to capitalize on that for digital annotations. The three types of annotations I have implemented in the system differ in what they are meant to be used for.

5.3.1 Anchoring to Documents

Anchoring to documents is equivalent to writing a note in the inside cover of a book. The annotation is meant as a guide for future readers of important or relevant information. In the current system it is possible to create annotations for particular documents but I have not worked out the details of where to place them so that they are visually prominent but not always in the user's way. A possible solution would be to place an information bar at the top of my document viewer with a pull-down list of annotations to the document.

5.3.2 Anchoring to Segments of Text

This is analogous to highlighting a section of text in a book. Your annotations stay locked to that position on that page in that document. The main difference in this type of annotation to physical books and shared digital documents is that the digital documents can be stored in a number of locations and can be changed frequently. The MVD project has tackled this issue with a number of smart anchoring algorithms [ANC].

5.3.3 Anchoring to Specific Terms

While the other two annotation types are useful for communication, I feel that anchoring to specific terms is a great contribution of this project. The concept is rather simple; an annotation of word X will replicate the annotation to every instance of term X in every document. This is useful for important terms that appear in many locations on many documents. A compelling example is biologists studying a disease that involves dealing

with many different genes and other biological elements. Since these entities interact with each other in many different ways, a particular gene could appear in numerous documents and all instances of the gene would convey the annotation made at one point. It is important to mention that this is not meant to be substitute for a notification system. It could potentially be used as a link building mechanism similar to the notion of used in Wikis.

5.4 Avoid Information Overload

Information overload is a common problem when working on complex issues and for group collaboration. The combination of the two together requires contextual and organizational filters to allow users to work effectively. I approach the former by allowing annotation authors to set priority levels and key words on their annotations and to give readers the ability to filter out content based on these. I use the two basic organizational units, person and group, as a filtering mechanism. Table 4 describes each of the filtering mechanisms and what they are used for. Figure 9 shows the UI windows that are used.

Property	Description	Values	Filter by	
			Author	Reader
Context	Based on the context that the annotation was written in and the specialty of the author	User defined		X
Priority	Urgency of the annotation	Urgent, Medium and None		X
Access	Controls who can see the annotation	Person, Group or both	X	
Authority	Determines who's annotations get displayed	Person, Group or both		X

Table 4 Features that can be filtered on

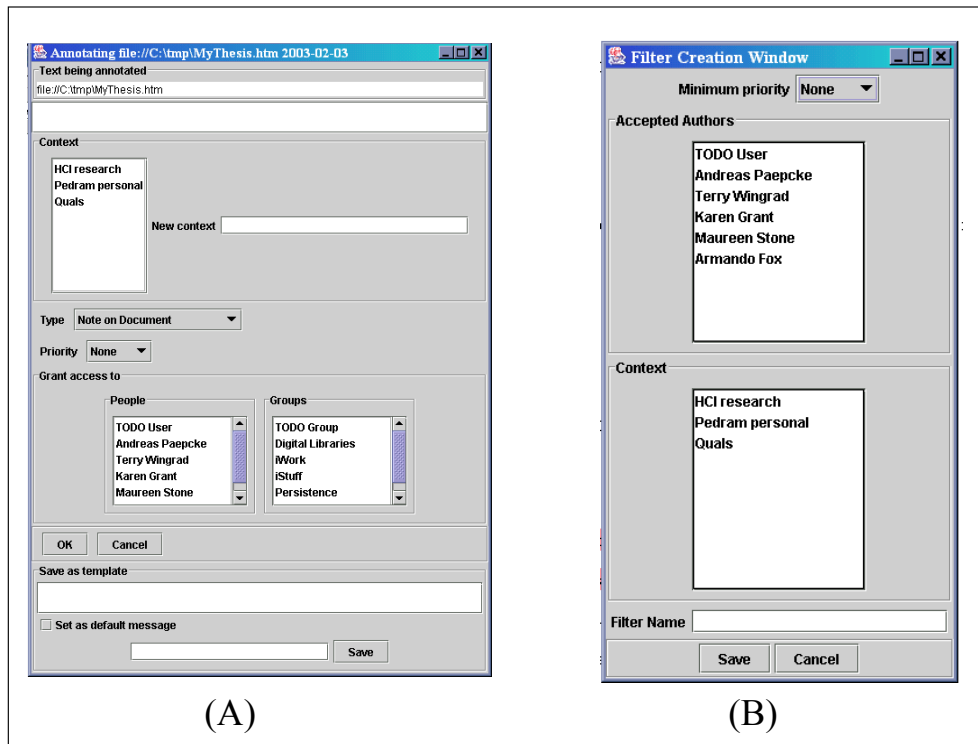


Figure9 (A) Annotation creation fields, (B) Filter fields

I will offer an example to show how these features can be used in a large group setting. This example revolves around the coaches, managers, physical therapists and doctors of a football team. I use an example from football because it is complex enough to demonstrate the concepts but simple enough for it to be well understood. In this example

three main job types contribute to the proper running of a football team. The example starts when a key player of the team injures his knee in practice. The doctor who examines him determines that his injuries are severe enough for him to miss the next two games. The doctor uses mRNA to update the player's medical report with an annotation. The note contains urgent information that needs to be viewed by the head coach, trainers and other team doctors. Because this is a common occurrence the doctor has a template annotation with the following values (Priority=Urgent, AllowAccess="Injury Watchers", Context="Injury", AnchorTo="Term"). Beforehand the doctor or an administrator of the system has created a group called "Injury Watchers" that consists of the head coaches, doctors and physical therapists. In the annotation the doctor can put information about the injury, necessary physical therapy and how long the player will be unable to play. Now this annotation is located everywhere that the users name is located, in this example there is a team roster and play book that coaches, therapists and doctor's view. When coaches look at their rosters or play books they will notice that there is an urgent note about the player, once they open the note they see that he is injured and that they will have to use an alternate player for the next two games. Physical therapists and other doctors will also see this note when looking at the team roster and thus prepare for treating the player's injury. In this example, writing just one annotation informs the relevant people in the organization while shielding people like managers and owners from information overloading. This same technique can be used to draw communication boundaries between offensive, defensive and special teams coaches.

The use of context is of extreme relevance to multi-disciplinary groups. Imagine a group of biologists and computer scientists working on a project to simulate cell

activity. In this case the burden of learning a new field falls on the computer scientists. In an effort to minimize the learning curve and offer the right level of abstraction the biologists could create a new context “Bio for CS people” and use it to annotate relevant terms. This is not meant to imply that biologists will use this system as their only means of communication with the computer scientists or that they must put detailed explanations in their notes. But they can use this system to support the learning effort and simply put links to websites about particular terms.

6 System Architecture

The mRNA system is based on an annotation server and a special web browser that can display annotations. The system is based on a client server architecture where the mRNA annotation server handles annotation requests from mRNA browsers that have been registered with it. My goal was to have a working model to demonstrate my concepts so while I will describe the entire mRNA system, some of the functionality was not fully implemented. In the “Future Work” section I will list these parts and give descriptions of possible improvements to the system.

The mRNA browser is responsible for displaying the documents, making requests to the annotation server, displaying annotations and updating the annotation server with new annotations. The mRNA annotation server is responsible for storing and serving annotations based on filters and access rights. Figure 10 shows the components of the mRNA system in terms of communication channels. In the following sub sections I will describe the components and their communication in more detail.

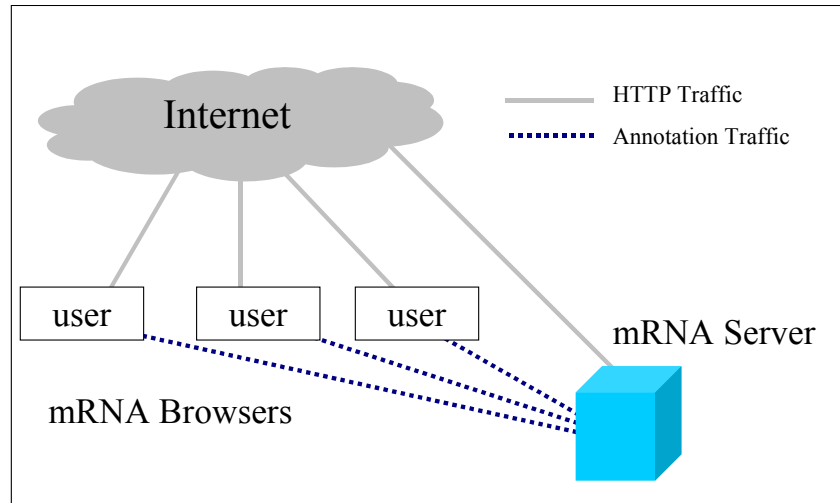


Figure 10 Parts of the mRNA system

6.1 mRNA Browser

The three basic functions of the mRNA browser are displaying documents, communicating with the annotation server and displaying annotations. It makes sense to talk about each of these functions in terms of reading and writing annotations.

A basic usage scenario is when a user views a document that has been annotated. The first step is for the user to type the URL of the document into the browser. At this point the browser does two things; it first accesses the remote page to display and it contacts the annotation server to get annotation information for the page. I will defer discussion of what the annotation server does until the next section. Once the browser has these two components it displays the annotation markers on top of the original document. The annotation layer, as shown in figure 11, also serves as a functionally component to handle user events and display the body of the annotation when they are selected. The MVD document model inspired this layered approach to annotations where layers represent different types of functionality.

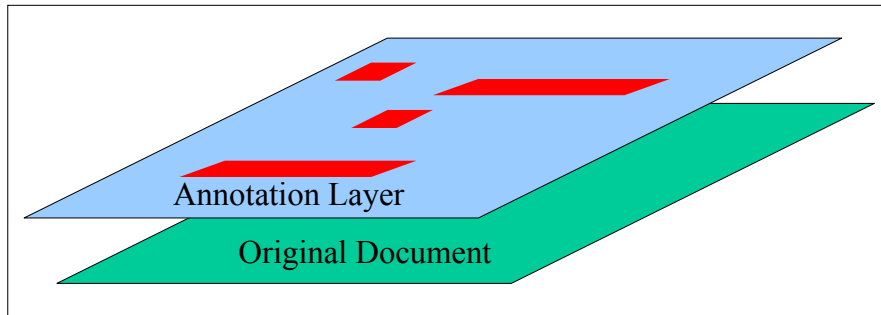


Figure 11 Annotation layer

6.2 mRNA Annotation Server

The mRNA annotation server stores and handles annotation events. In my research the highest priority was to develop the annotation event handling to show that the system works. While the design of the storage component is complete, work still needs to be done to handle data persistence between sessions. In its current incarnation the server can only handle persistence within a session; future work should focus on storing intersession information.

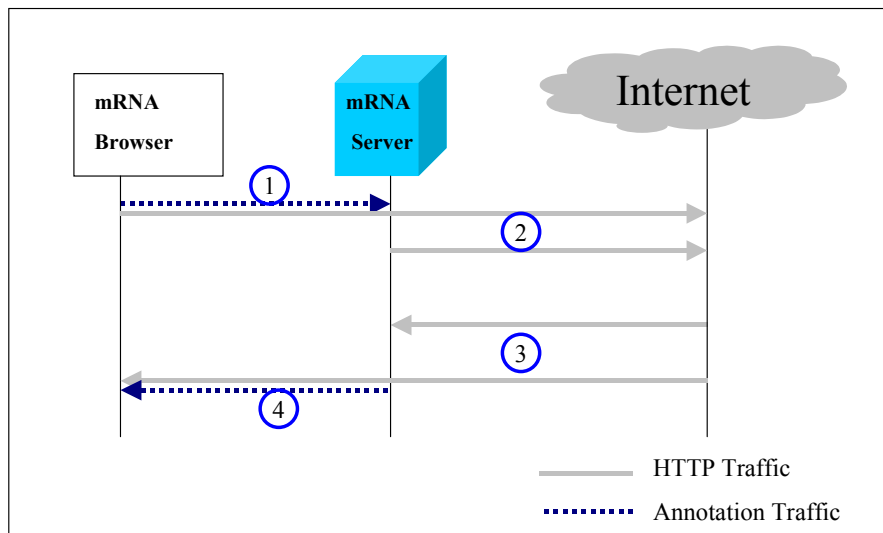


Figure 12 Page request sequence

Figure 12 shows a timing diagram of what steps occur when a page is requested. When an mRNA browser requests a web page the request is also relayed to the annotation server. When the server receives the request it analyzes the text of the page, filters annotations and then returns a list of annotations to the browser. The page is analyzed for specific terms and segments of text that have annotations associated with them. The filtering process that was discussed in section 5.4 is handled in the annotation server. The server first determines what annotations the reader is authorized to read. It then uses the reader specific filter to return annotations that have the correct priority, author and context information.

The entire system was written in Java because of the many capabilities that it provides. Ease of GUI construction, displaying web pages, database connectivity and communication mechanism are among the key benefits of using Java.

7 Conclusion

I will conclude by focusing on the three areas I learned most about during this research project: successful annotation architectures, the role of annotations to support group collaboration and annotations as a communication medium. I will also give a short list of lessons I learned about HCI research during the course of my RAship.

7.1 Annotation Architectures

The myriad of operating systems, data standards and applications makes any annotation system doomed to isolation and eventual death unless it conforms to an open standard. Standards such as RDF or CAF must become widely adopted so that collaboration and productivity software can become commonplace for all domains where

group work is necessary. This is necessary because of the huge cost of developing specialized software for different domains. Without standardized ways of modeling complex data, we will be overwhelmed by integration efforts every time a new software application comes to market. When one open annotation standard is widely adopted, collaborative annotation systems will be as widespread as text-editing software. This is just as important to researchers as it is to developers making consumer products for the simple fact that effort spent on low-level design takes away from meaningful research.

This principle applies to my work on mRNA also. Much of my time was devoted to coming up with a data model for how the different concepts (user, workgroup, annotation, filter, broker, etc) should be modeled and incorporated into the architecture. The issue becomes complicated when data persistence becomes a concern. This latter problem can be partially solved by using Java Enterprise Java Beans (EJB) to automatically handle database representation and persistence. Annotation data interoperability is another issue that needs to be addressed but is beyond the scope of this research.

7.2 Annotations in Supporting Group Collaboration

Reading about the related CSCW projects it became very clear that the successful projects have been based on software that consumers are familiar with. In particular the Microsoft annotation work and the Wiki concept rely on the user's familiarity with web browsers. Both of these projects are in wide spread use, while the MVD project, which has created many great ideas, has not been widely adopted. I am not claiming that this is the only reason why these projects took different paths, but am confident that it has played a part in the real world application of certain ideas over others.

In creating the mRNA browser I wanted to model it after the web browser design because it is basically a web browser with additional functionality. It falls short of providing all of the functionality of a traditional web browser and that can limit its success in being used. In the “Future Work” section I will describe some ways this can be overcome.

7.3 Annotations as a communication Medium

Using annotations as a communication medium in CSCW has many facets to understand. A shared communication space is effective for small or narrowly focused groups, as noted in the Wiki work of section 4.1.1. This seems to logically extend to the domain of annotations. The Microsoft field studies were targeted at large groups that were very focused on one thing, software development.

In order to be a reliable communication medium, annotations must be supported with a notification mechanism. The example of section 5.4 only worked because the members of the group constantly used the team roster. If this had not been the case, important messages would go unnoticed which would result in potential problems. I will discuss this further in the future work section.

7.4 Lessons Learned About HCI Research

In the course of working on this project I have learned many things about HCI research and research in general. It differs very much from systems research, which I am familiar with, in one fundamental way; user studies. From the beginning of the project you must look ahead to think how to test concepts on a set of users and what type of data is meaningful. This also means that your rapid-prototype must be something that can either

be tested in a short period of time in a controlled setting or be deployed to collect statistical data. It is also important to select an audience that exhibits similar traits to your target audience, or if possible, to try it on your target audience.

There is a catch 22 of creating a GUI: you can't really understand how it is going to be used until you create it but you can not really understand what it should look like until it is used. I dealt with this by first producing a software specification for the system design and an initial design of what each screen and menu would look like. This proved useful because I was able to test if I could get certain tasks done with the menus I had created. This allowed me to catch a number of problems that would have been time-consuming to change once the code was already done.

8 Future Work

I think the temporal aspect of annotations is an important research topic. Of particular interest to me is using statistical information about usage patterns to determine how long to present an annotation. This becomes an important problem if you think of projects that can take months to years and the buildup of annotations could potentially clutter up any document. This is actually low hanging fruit because Microsoft has already performed many long term studies of annotations and have temporal data that can be analyzed [NOT, AAC]. The difficulty is in acquiring that data from Microsoft.

Another interesting avenue to explore would be to deploy a wiki into a multi-disciplinary research project. The study would be focused on seeing how group dynamics affect reading and writing of such a malleable shared medium. Because the source is freely available and written in Perl it would require little work to add a few additional modules to track usage statistics beyond the basic module already available.

We might be able to observe some patterns that help in designing the next version of mRNA.

On the engineering aspects there are many things that could be done to extend and improve the current system so that it could be used in an initial user study. A major improvement would be to implement the mRNA browser functionality into Mozilla or Internet Explorer as a plugins. This would build on the widespread use of such applications and the APIs and packages to do this already exist [PLU]. Another improvement would be to port the mRNA server code into a J2EE application server and use EJBs to model the relationship between components in the system. This would be beneficial because it would automatically employ a database to do the data modeling, storage and consistency constraints. Another aspect of the project that needs development is an administrative tool to setup groups and users on the system with the proper access control. I did not address this because it was the lowest of the priorities, but relying on a dedicated administrator of the system does not scale beyond short-lived, small experiments.

In the current system, important annotations can go unnoticed because there is no way to notify users of new annotations that may be of interest to them. Adding notifications to this system is important because it extends the use of annotations beyond just a collective memory to a true communication medium. Microsoft has already shown a base implementation with “sideshow”. The key task would be to develop a scalable architecture for such a system.

A Note About the Name mRNA

During my recent internship at Agilent research labs I was exposed to many concepts in biology. One of the most intriguing concepts was of Messenger RNA (mRNA). The interesting thing about mRNA is that it is transcribed from DNA and synthesizes all the protein in the human body.

Acknowledgements

After completing this work I look back at all the obstacles overcome and lessons learned. Accompanying each of these obstacles and lessons I see the faces of the people who helped me meet the obstacles and learn the valuable lessons. The two most crucial people in this process are my advisors Terry Winograd and Andreas Paepcke. It is only through their help, support and constant feedback that made it possible for me to complete this work. I would also like to thank Annette Adler, Robert Kincaid, Allan Kuchinsky, David Moh and Aditya Vailaya from the Life Sciences Technologies Laboratory at Agilent Technologies. They provided the initial motivation for the research as well as the funding that made this work possible. Along with supporting this work they brought me on for a summer internship where I worked on an interesting problem in bioinformatics. During this internship Aditya served as my mentor where he taught me a great deal about how to manage time, focus and energy as a researcher. These are some of the most valuable lessons I learned during this project and I will continue to practice them well into my future.

Bibliography

- [COM] M. Röscheisen, C. Mogensen, and T. Winograd. "Shared web annotations as a platform for third-party value-added information providers: Architecture, protocols, and usage examples". Stanford Integrated Digital Library Project STAN-CS-TR-97-1582
- [MVD] T.A. Phelps and R. Wilensky. "Multivalent Annotations". In Proc. European Conference on Research and Advanced Technology for Digital Libraries, 1997
- [NOT] A. J. Bernheim Brush, D. Barger, J. Grudin, A. Gupta." Notification for Shared Annotation of Digital Documents". CHI 2002: 89-96
- [CAF] D. Barger and A. Gupta. "A Common Annotation Framework". Microsoft Tech. Report MSR-TR-2001-108.
- [AAC] J. Cadiz, A. Gupta, and J. Grudin. "Using Web Annotations for Asynchronous Collaboration Around Documents". Proceedings of CSCW 2000, 309-318.
- [ECO] C.C. Marshall, "Toward an Ecology of Hypertext Annotation". Proceedings of HyperText 1998, 40-48.
- [MAR] C. Marshall. "[The Future of Annotation in a Digital \(Paper\) World](#)". In *Successes and Failures of Digital Libraries* (Harum and Twidale, eds). Urbana-Champaign: University of Illinois, 2000, pp. 97-117.

- [MDL] C. Marshall. "[Annotation: from paper books to the digital library](#)" in *Proceedings of the ACM Digital Libraries '97 Conference*, Philadelphia, PA (July 23-26, 1997).
- [SYS] S. Cousins, M. Baldonado, and A. Paepcke. "A Systems View of Annotations". *Xerox PARC Tech Report P9910022*
- [WIK] B. Leuf and W. Cunningham, [The Wiki Way. Quick Collaboration on the Web](#), Addison-Wesley, 2001
- [CSC] J.C. Grundy, J.G. Hoskin and W.B. Mugridge. "Low-level and High-level CSCW Support in the Serendipity Process Modelling Environment". In *Proceedings of OZVHI 1996*
- [SER] J.C. Grundy, J.G. Hosking and W.B. Mugridge, "Coordinating distributed software development projects with integrated process modelling and enactment environments". *Proc. of 7th IEEE Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, Stanford, CA (June 1998).
- [KRR] R. Davis, H. Shrobe and P. Szolovits. "What is Knowledge Representation?". *AI Magazine*, 14(1):17-33. 1993.
- [ANC] T.A. Phelps and R. Wilensky. "Robust Intradocument Locations", *Proceedings of the Ninth World Wide Web Conference*, 15-18 May 2000, Amsterdam.
- [BNS] R. Kincaid. "BNS: A DNS inspired bimolecule naming service"
- [W3C] <http://www.w3.org/Collaboration/>
- [DAS] <http://biodas.org/>

[ZOP] <http://zwiki.org/FrontPage>

[TWI] <http://twiki.org/>

[PLU] <http://www.mozilla.org/projects/blackwood/java-plugins/>